

Interactive Ruby or irb is an interactive programming environment that comes with Ruby. It was written by Keiju Ishitsuka.

Usage Syntax:

To invoke it, type irb at a shell or command prompt, and begin entering Ruby statements and expressions. Use *exit* or *quit* to exit *irb*.

```
$ irb[.rb] [options] [programfile] [arguments]
```

Here is a complete list of options:

SN	Command with Description
1	-f Suppress reading of the file <code>~/.irbrc</code> .
2	-m bc mode <i>loadmathnlibrarysofractionsormatrixareavailable</i> .
3	-d Set \$DEBUG to true (same as ruby -d).
4	-r load-module Same as ruby -r.
5	-I path Specify \$LOAD_PATH directory.
6	--inspect Use inspect for output <i>defaultexceptforbcmode</i> .
7	--noinspect Don't use inspect for output.
8	--readline Use Readline extension module.
9	--noreadline Don't use Readline extension module.
10	--prompt prompt-mode – – <i>prompt – modeprompt – mode</i>

Switch prompt mode. Predefined prompt modes are *default*, *simple*, *xmp*, and *inf-ruby*.

11 **--inf-ruby-mode**

Use prompt appropriate for *inf-ruby-mode* on Emacs. Suppresses *--readline*.

12 **--simple-prompt**

Simple prompt mode.

13 **--noprompt**

No prompt mode.

14 **--tracer**

Display trace for each execution of commands.

15 **--back-trace-limit n**

Display backtrace top n and tail n. The default value is 16.

16 **--irb_debug n**

Set internal debug level to n *notforpopularuse*.

17 **-v --version.**

Print the version of irb.

Example:

Here is a sample of irb evaluating a variety of expressions::

```
$ irb
irb(main):001:0> 23 + 27
=> 50
irb(main):002:0> 50 - 23
=> 27
irb(main):003:0> 10 * 5
=> 50
irb(main):004:0> 10**5
=> 100000
irb(main):006:0> x = 1
=> 1
irb(main):007:0> x + 59
=> 60
irb(main):005:0> 50 / 5
=> 10
irb(main):008:0> hi = "Hello, Mac!"
=> "Hello, Mac!"
```

You can also invoke a single program with *irb*. After running the program, *irb* exits. Let's call our *hello.rb* program:

```
$ irb hello.rb
hello.rb(main):001:0> #!/usr/bin/env ruby
hello.rb(main):002:0*
hello.rb(main):003:0* class Hello
```

```
hello.rb(main):004:1> def initialize( hello )
hello.rb(main):005:2> @hello = hello
hello.rb(main):006:2> end
hello.rb(main):007:1> def hello
hello.rb(main):008:2> @hello
hello.rb(main):009:2> end
hello.rb(main):010:1> end
=> nil
hello.rb(main):011:0>
hello.rb(main):012:0* salute = Hello.new( "Hello, Mac!" )
=> #<Hello:0x319f20 @hello="Hello, Mac!">
hello.rb(main):013:0> puts salute.hello
Hello, Mac!
=> nil
hello.rb(main):014:0> $
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js